

**In the Claims:****Claim 1 (Currently Amended)**

- 1 1. For use in a data processing system having a memory coupled to multiple  
2 requesters, a memory coherency system, comprising:  
3 a memory circuit coupled to provide a copy of requested data from the  
4 memory to a first requester, and to initiate invalidation operations to invalidate all  
5 read-only copies of the requested data that are stored by one or more other  
6 requesters, wherein the data is provided before the invalidation operations are  
7 completed; and  
8 a circuit included within the first requester and responsively coupled to the  
9 memory circuit to execute an instruction that causes the first requester to  
10 temporarily enter a stalled state until all of the invalidation operations have been  
11 completed.

**Claim 2 (Cancelled)****Claim 3 (Original)**

- 1 3. The system of Claim 2, wherein the memory circuit includes a request  
2 channel and a response channel.

**Claim 4 (Original)**

- 1 4. The system of Claim 3, wherein the memory circuit includes an  
2 acknowledge tracker to initiate the transfer of an acknowledge to the first  
3 requester when all invalidation operations for the requested data are completed.

**Claim 5 (Original)**

- 1 5. The system of Claim 4, wherein the requester includes a request tracking  
2 circuit responsively coupled to the memory to record when the acknowledge is  
3 outstanding for the requested data.

**Claim 6 (Original)**

- 1 6. The system of Claim 1, wherein the first requester issues multiple  
2 requests, and wherein the circuit prevents any further instruction processing from  
3 occurring within the first requester until all invalidation operations have been  
4 completed for all of the multiple requests.

**Claim 7 (Original)**

- 1 7. The system of Claim 6, wherein the first requester is a processing node  
2 that includes multiple processors, wherein the circuit resides within one of the  
3 multiple processors and includes logic to execute an instruction to stall the  
4 processor until all of the invalidation operations have been completed for data  
5 previously provided to the processor.

**Claim 8 (Original)**

- 1 8. The system of Claim 6, wherein the first requester is a processing node  
2 that includes multiple processors, wherein the circuit resides within one of the  
3 multiple processors and includes logic to execute an instruction to stall the  
4 processor until all of the invalidation operations have been completed for data  
5 previously provided to predetermined ones of the processor in the processing  
6 node.

**Claim 9 (Original)**

- 1 9. For use in a system having multiple requesters coupled to a shared  
2 memory, a method for controlling processing of requests, comprising:  
3 a.) issuing a request for data by a requester to the shared memory;

- 4           b.) providing the data from the shared memory in response to the request  
5 before all read-only copies of the data retained by other requesters have been  
6 invalidated; and  
7           c.) stalling the requester until all of the read-only copies have been  
8 invalidated.

**Claim 10 (Original)**

- 1 10. The method of Claim 9, wherein step c.) includes the initiation by the  
2 requester of a hardware sequence to stall the requester until the read-only copies  
3 have been invalidated.

**Claim 11 (Original)**

- 1 11. The method of Claim 10, wherein the requester is an instruction  
2 processor, and further including execution of a predetermined instruction to  
3 initiate the hardware sequence.

**Claim 12 (Original)**

- 1 12. The method of Claim 11, wherein the instruction is part of the hardware  
2 instruction set of the instruction processor.

**Claim 13 (Original)**

- 1 13. The method of Claim 9, and including repeating steps a.) and b.) for  
2 multiple requests, and stalling the requester until all read-only copies of any data  
3 requested by any of the multiple requests have been invalidated.

**Claim 14 (Original)**

- 1 14. The method of Claim 9, wherein the requester is a processing node  
2 containing multiple processors, and wherein the method comprises:  
3           a.) issuing a request for data by one of the processors to the shared  
4 memory;

- 5           b.) providing the data from the shared memory to the processor in  
6 response to the request before all read-only copies of the data retained by other  
7 requesters have been invalidated; and  
8           c.) stalling the processor until all of the read-only copies have been  
9 invalidated.

**Claim 15 (Original)**

- 1   15.   The method of Claim 14, wherein a.) and b.) are repeated for multiple  
2 requests, and wherein step c.) includes stalling the processor until all read-only  
3 copies of any data previously provided to the processor have been invalidated.

**Claim 16 (Original)**

- 1   16.   The method of Claim 15, wherein steps a.) through c.) may be performed  
2 for more than one processor in the processing node, and wherein step c.)  
3 comprises stalling a processor until all read-only copies of any data previously  
4 provided to the processor have been invalidated.

**Claim 17 (Original)**

- 1   17.   The method of Claim 15, wherein steps a.) through c.) may be performed  
2 for more than one processor in the processing node, and wherein step c.)  
3 comprises stalling a processor until all read-only copies of any data previously  
4 provided to predetermined ones of the processors in the processing node have  
5 been invalidated.

**Claim 18 (Original)**

- 1   18.   The method of Claim 9, and further comprising issuing an inter-processor  
2 interrupt by the requester to another requester to indicate that data stored within  
3 the shared memory by the requester may be accessed by the other requester.

**Claim 19 (Original)**

- 1 19. A system for use in managing requests within a data processing system,  
2 comprising:  
3 means for providing data in response to a request before all read-only  
4 copies of the data that reside within the data processing system at the time of  
5 receipt of the request have been invalidated; and  
6 means for selectively discontinuing predetermined data processing tasks  
7 until all of the read-only copies have been invalidated.

**Claim 20 (Original)**

- 1 20. The system of Claim 19, wherein the data processing system includes a  
2 shared main memory coupled to multiple instruction processors, and wherein the  
3 means for selectively stalling includes means provided within at least one of the  
4 instruction processors for executing a predetermined instruction to stall the at  
5 least one instruction processor.

**Claim 21 (Original)**

- 1 21. The system of Claim 20, wherein the means for executing includes means  
2 for stalling the respective instruction processor until all read-only copies of any  
3 data that was previously requested by the instruction processor have been  
4 invalidated.

**Claim 22 (Original)**

- 1 22. The system of Claim 21, wherein execution of the predetermined  
2 instruction by a instruction processor issues a request for data, and wherein the  
3 means for selectively stalling includes request tracking means for delaying return  
4 of the data until all read-only copies of any data that was previously requested by  
5 the instruction processor have been invalidated.